

AFRL-IF-RS-TR-2003-43
Final Technical Report
March 2003



KNOWLEDGE BASED COLLABORATION WEBS

Massachusetts Institute of Technology

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2003-43 has been reviewed and is approved for publication.

A handwritten signature in black ink, appearing to read 'Edward Depalma', with a long horizontal flourish extending to the right.

APPROVED:

EDWARD DEPALMA
Project Engineer

A handwritten signature in black ink, appearing to read 'James W. Cusack', with a large loop at the beginning and a horizontal flourish at the end.

FOR THE DIRECTOR:

JAMES W. CUSACK, Chief
Information Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 074-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE MARCH 2003		3. REPORT TYPE AND DATES COVERED Final Jun 97 – Dec 00
4. TITLE AND SUBTITLE KNOWLEDGE BASED COLLABORATION WEBS			5. FUNDING NUMBERS C - F30602-97-2-0239 PE - 62301E PR - F302 TA - 01 WU - 00	
6. AUTHOR(S) Howard Shrobe				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Massachusetts Institute of Technology Artificial Intelligence Laboratory Cambridge Massachusetts 02139			8. PERFORMING ORGANIZATION REPORT NUMBER N/A	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/IFSF 525 Brooks Road Rome New York 13441-4505			10. SPONSORING / MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-2003-43	
11. SUPPLEMENTARY NOTES AFRL Project Engineer: Edward DePalma/IFSF/(315) 330-3069/ Edward.DePalma@rl.af.mil				
12a. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.				12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 Words) The Knowledge Based Collaboration Web (KBCW) project at the MIT Artificial Intelligence Laboratory (AI) exploited representations and techniques used in AI for research and development of a platform and tools to support collaboration. A KBCW consists of nodes and links, where nodes represent a document, or a fragment of a document with a coherent prepositional content and links represent relationships between the nodes. Both the nodes and links are made accessible to clients via its Universal Resource Name (URN) which serves as a location independent identifier for the World Wide Web. A KBCW represents a set of logical statements summarizing the content of the documents as well as the relationships among the contents of the documents. The particular focus of this project was on face-to-face and remote collaborative processes in the creation of knowledge products, like software and military security intelligence. The scope included tools for a) managing the collaborative interactions, b) representing parts and relationships in the cumulative knowledge product, and c) the enhancement of smart spaces/intelligent rooms for support of collaborative meetings and capture of contents.				
14. SUBJECT TERMS Knowledge Base, Universal Resource Name, Intelligent Rooms, Collaboration, Artificial Intelligence, Smart Spaces			15. NUMBER OF PAGES 50	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

Table of Contents

Introduction	1
Section 1: Guiding Insights	1
Section 2: Achievements	3
Reflective Group Interaction Mediation:	4
Evaluation:	4
Resource Management:	4
Using Natural Language Content to Facilitate Group Interactions:.....	5
Intelligent Room Technology for Natural Interaction:.....	6
Substrate Technology for Broad Area Interaction:.....	6
Demonstration of Collaborative Design:	7
Section 3: The Intelligence Analysis System	8
Background	8
The Scenario	9
Bayesian Processing	15
The Planning Phase	17
Technologies used in the scenario	18
Section 4: Transition to Project Oxygen	21
The Meeting Manager	22
Sketch Understanding	27
Service Mapping and Resource Management.....	29
Section 5: The Importance of URN	32
What's Wrong With Uniform Resource Locators (URLs)?	32
Transport Protocol Specificity	32
Location Specificity.....	32
Mobility Limitations	32
Version Omission	33
Fragment References.....	33
How Uniform Resource Names (URN) Address the Shortcomings Of URLs ..	34
New Opportunities Opened by URNs.....	35
URN Research During The KBCW Project.....	39
Conclusions.....	41
Section 6: Project Outreach	44
Cumulative List of Papers Presented at Meetings.....	44
Consultative and Advisory Functions to Other Laboratories.....	45

Table of Figures

Figure 1: KBCW Contents.....	10
Figure 2: KBCW Contents.....	12
Figure 3: Plan Structure	13
Figure 4: URN	14
Figure 5: Inadequate Discussion	15
Figure 6: Bayesian Network.....	16
Figure 7: ‘Logical and’ Bayesian Network	17
Figure 8: Project Management Tools	19
Figure 9: Project Management Tools	19
Figure 10: Project Management Tools	20
Figure 11: Participants view a commitment being made in a prior meeting	23
Figure 12: A view of the current meetings Agenda	24
Figure 13: The KBCW web structure used by the Meeting Manager	25
Figure 14: The displays maintained by the Meeting Manager.....	26
Figure 15: The Drawing System Captures and Cleans Strokes.....	28
Figure 16: The Sketch Understanding System Creates a Simulation	28
Figure 17: Services are mapped to plans. Plans are evaluated by their cost- benefit.....	31

Introduction

This is the final report on the Knowledge Based Collaboration Web (KBCW) project at the MIT Artificial Intelligence Laboratory, June 12, 1997 – December 31, 2000. The project aimed to exploit representations and techniques used in AI for research and development of a platform and tools to support collaboration. The particular focus was on face-to-face and remote collaborative processes in the creation of knowledge products, like software and military/ security intelligence. The scope included tools for a) managing the collaborative interactions, b) representing parts and relationships in the cumulative knowledge product, c) the enhancement of smart spaces/ intelligent rooms for support of collaborative meetings and capture of contents.

The report is divided into several sections:

1. A review of the KBCW guiding insights;
2. A review of the project's achievements;
3. A description of the deployment the tools produced in a defense intelligence analysis scenario;
4. A review of the KBCW cross connections with other projects at the AI Laboratory and its anticipation of collaboration in an Oxygen environment;
5. An argument for the Universal Resource Name (URN) system developed as part of the project;
6. A summary of professional outreach activities motivated by the project.

An important part of KBCW was its training of graduate students. Their research and results are noted in the main body of this report; significant parts of their theses based on their work are presented in the appendix. Without the students, KBCW would have accomplished much less. Their work dealt with problems that arose in the pursuit of KBCW goals, problems of meeting facilitation and support, resource management, knowledge representation, specifically annotation, and interface and presentation design. In some cases, they also crossed project boundaries to enhance work on the human-computer interface being done in the START and Intelligent Room projects, which were active at the AI Laboratory at the same time as our project.

Section 1: Guiding Insights

The central insight guiding our work: In domains like software development and (military/ political) intelligence assessment domains, collaboration is motivated by a shared need to solve a common problem and enabled by joining shared and individual knowledge and understanding of the problem domain. The process

itself is information driven, opportunistic, and evolutionary; each step taken depends on the information already developed and the capabilities, interests and workloads of currently available personnel.

No single workflow model can guide collaboration for all problems and across all sets of resources and personnel. How the collaborators interact depends on where they are in the solution process. In early stages of problem solving, brainstorming and exploration of many alternatives are appropriate, but in later stages, convergence is preferable -- participants need to keep a common focus and not get diverted by new ideas. The collaboration support system must accordingly adapt its style of interaction management. In addition, because information plays so critical a role, the support system must facilitate access to the richness of common and personal knowledge bases. Common knowledge should not be reduced to what can be carried by shared whiteboards; information search and discovery should not be limited to what can be expressed in standard interfaces.

On this view, we saw that collaborative problem solving would be significantly enhanced by a support system that understood:

1. the content of the (current) problem solving task being supported;
2. the problem solving context of the task being undertaken (i.e., where it fits in the overall solution);
3. the organizational context of the participants in the collaboration.

These capabilities respectively allow the system to provide significant help with the task at hand, manage group interactions in ways appropriate to the task, and marshal the needed human and organizational resources. An essential key to achieving these capabilities is representing knowledge about the problem domain, problem solving processes, group interactions and organizational resources. Achieving these goals requires the system to provide:

- A framework that can assimilate the specific knowledge and information relevant to the domain and organization;
- Evolving representations of the problem solving process and the partial solutions;
- Software agents to monitor the process and note opportunities for engaging humans and others agents;
- Interfaces based on natural language processing and machine vision technologies to enable human interaction with the system and capture human outputs.

In brief, the horizon for the KBCW project was a knowledge based system that could direct and support collaborations for complex problem solving – collaborations where large groups of interacting human and software agents

dynamically (re) arrange themselves in appropriate teams for the emergent sub-problems.

To pursue these goals and subgoals, we intended to build as follows on several technologies, which members of KBCW had previously developed:

1. Open Meeting server, a platform for large scale, multilateral, asynchronous stylized discussions. This server enabled and regulated the attachment of comments, queries, etc., by discussants to other discussants' comments and queries, according to a specific argument grammar. It would be extended to support various conversational processes, such as brainstorming, which could be applied at appropriate phases in the collaboration.
2. White House Electronic Documents Server, a text server system with capabilities of automatically categorizing and indexing input texts and distributing them to mailing lists created on the fly. Extend to support distribution of multi-media documents or fragments thereof according to potential collaborators interests, roles, expertise, security clearance, downtime or other arbitrary attribute in their profiles, relevant to potential contribution to the problem solving.
3. START a system for acquisition and semantic understanding of textual information. START parses natural language queries and returns selections from its text base in response to information sought through these questions. START would be extended to support annotation of its textbase, in particular automatically generated summaries of input that would eliminate the need for often problematic full text parsing.
4. The Intelligent Room – a smart space with an array of agent based, user responsive tools for multimedia display of database information and capture of events/ interactions in the room. The intention was to integrate these tools to record collaborative sessions, particularly decisions and commitments made.

Section 2: Achievements

Our actual work on KBCW was organized into 6 major areas, several of which had discrete sub-areas:

1. Reflective Group Interaction Mediation
2. Using Natural Language Content to Facilitate Group Interactions
3. Intelligent Room Technology for Natural Interaction
4. Substrate Technology for Broad Area Interactions
5. Demonstration of Collaborative Intelligence (an Application)
6. Demonstration of Collaborative Design (an Application)

The research and results for each area are summarized below and described more extensively in subsequent sections of this report.

Reflective Group Interaction Mediation:

We researched and implemented a mechanism for the representation and retrieval of organization goals and plans, and implemented an interactive editor for plans for collaborations occurring within their contexts. The editor is supported by a database of techniques for mediating group interactions, appropriate to the various stages of the collaboration. (Prototypes were demonstrated at DARPA and Rome Labs, March 1998.) These facilities were extended with mechanisms for executing and monitoring collaborative processes, e.g., calling meetings, recording commitments, and tracking the collaboration according to plan. We also provided decision theoretic substrate to meet two broad concerns: evaluation of the assertions and proposals by participants, and management of resources in support of the collaboration.

Evaluation:

Particularly in the early stages of collaboration in our application domains, many interactions involve presentation of claims, viz., evidence and arguments, for (and against) different hypotheses or proposals. By having participants to associate conditional probabilities to their claims, the interaction management system can dispassionately assess the likelihood of hypotheses/ proposals as complex argument and evidence chains are created through multilateral communication. When certain thresholds are crossed, it can call for new collaborative steps. We provided for such use of probabilistic decision theory by incorporating basic Bayesian network algorithms into our system. This use is further described in the section on the demonstration of the system for intelligence analysis.

Resource Management:

We implemented a system that uses decision theoretic techniques to decide which resources to apply to a group interaction. This system structures interactions into a multi-layered taxonomy of abstract services. The abstract services are in turn rendered by more concrete services until the implementation level is reached. For example, notification service can be each achieved by beeper, remote screen access (popping up a message), by faxing or by printing on a printer in the user's office, or by email. Each of these has an associated cost and an associated value for each of several properties such as speed. The system must assess how much value it places on each of these properties (e.g. speed be of minor importance, guaranteed delivery may matter a lot). Given this the system conducts a best first search to locate the rendition of the abstract service which maximizes the ratio of expected benefit to expected cost. A category of abstract services of particular relevance to collaboration support is

that of the meeting service, which has both synchronous and asynchronous specializations.

There was a subsequent implementation of these ideas to manage the services available within our Intelligent Room. This implementation understands that separate activities within the room may contend for the same resources (e.g. screen space, or use of a video projector), the priorities of the activities are dynamically changing, and seizing an asset from its current users has a high social penalty. The new implementation accounts for all these factors in making an allocation decision, then actually implements the allocation of the service and manages the setup of appropriate sockets and agents. (This is described more fully in "Transition to Oxygen," the section describing cross-support between KBCW and other AI Laboratory projects.)

Using Natural Language Content to Facilitate Group Interactions:

We completed three major tasks in this area. We integrated our representations of natural language interactions, vis a vis the structural roles of the utterances or comments with our representations in our document management system vis a vis the content categories and diexis of the documents. Hence, documents and transcriptions can be supported by the same reference (naming), indexing and annotation systems, allowing support documents to be seamlessly introduced in (representations of) a collaboration. Second, we developed a forward chaining inference system capable of responding to both coarse grained representations of the collaboration web, i.e., the structural roles of the contributed information, as well as to the fine grained representations produced by the natural language system, i.e., the semantic understanding of the information. Third, we completed (in a Master of Engineering thesis) an automatic text summarization tool that can interact well with our START NLP system.

This last tool enables the document management system to index long technical documents that START cannot parse with short summaries that START can parse. The tool works by selecting a set of sentences likely to be good annotations in the sense of capturing the document's essence. Both structural and statistical clues are used to guide the selection. The tool was tested on a variety of textual genres and in most cases produced summaries that were judged comparable to those produced by hand.

Work was also began on a new system that will use information extraction techniques and "robust parsing" (i.e. parsers that keep going even when they can't parse a part of a sentence) as well as the structural and statistical clues in attempts to automatically index larger documents.

Intelligent Room Technology for Natural Interaction:

We designed and implemented a control system (named Meta-Glue) for the Intelligent Room resources. We integrated agents into this system that can support complex levels of interaction, e.g., an agent that can locate and replay a designated “significant event” in a video of a previous meeting. We completed (in a Master of Engineering thesis) a reimplementaion of Meta-Glue that has service mapping and decision theoretic (“business practice”) substrates. This new implementation takes into account possible contention, changing needs and social costs, when allocating resources among concurrent activities in the Intelligent Room. The KBCW project also replicated the facilities on the Intelligent Room in another office and brought them into daily robust use. (Facilities are more fully described in the “Transition to Oxygen” section.)

Substrate Technology for Broad Area Interaction:

Work in this area involved continuing enhancements of our Comlink and Open Meeting technologies. During the contract period we

- Extended Comlink's ability to generate Java, JavaScript and advanced HTML;
- Verified this improves the power of the system through sustained production use by professional document analysts;
- Created a role and task based access control framework for Comlink;
- Integrated the needed cryptographic support mechanisms;
- Expanded the set of inter-document links supported by Open Meeting to include connections among parts of plans and also stages of planning and implementation;
- Implemented an automatic categorization system for documents (per a designated set of categories) to enable intelligent routing to users per their interests, roles, etc.
- Enhanced web interfaces to support complex interactions with and via these servers.

In addition, we continued to enhance and support the use of these systems in projects at our lab and other research centers. In particular, we worked with the World Wide Web consortium and the IETF on the standardization of the Universal Resource Name (URN), a protocol and host independent unique identifier (The importance of URNs for network based collaboration is discussed in “Late Binding Identifiers.”) We fully implemented an URN resolver, made it available to the public and incorporated it into the White House publications system.

Demonstration of Collaborative Intelligence:

We developed a prototype environment for the collaborative interpretation of security intelligence. The motivating scenarios and notions of expertise and roles for inclusion in the collaboration drew on our experience in the HPKB

program. The system includes project management tools, resource (including personnel) description tools, natural language understanding facilities and sophisticated reasoning capabilities. A critical component of this facility is its ability to assess and combine probabilistic evidence about the events being interpreted. We built this capability by integrating Bayesian inference algorithms into our Comlink infrastructure so that our system can assess the strength or weakness of a hypothesis in terms of the argument structure build by the collaborating analysts. (This work was completed in a Master's of Engineering Thesis.) In September 2000, we tested the system with ten participants. See "The Intelligence Analysis System" section for a full description of the scenario used and the system capabilities.

Demonstration of Collaborative Design:

In this area, the domain of software design was used to focus effort to develop a collaboration platform and integrate appropriate technologies. A Comlink-style substrate was developed that was suitable for collaborative software development and annotation. This environment supports the management of individual software modules as discrete entities, supporting first class links. This enables their annotation and the collection of annotations about such entities across module boundaries. A Remote Method Invocation facility was also developed to allow remote clients written in Java to talk to a central server (Comlink) written in LISP. The resulting system has been used to structure a large software system, CL-http – the Comlink web server.

The work was completed in two Master of Engineering theses, particularly the Vincent thesis (see abstract below).

Section 3: The Intelligence Analysis System

Background

A knowledge based collaboration web (KBCW) consists of nodes and links. The nodes represent a document, or a fragment of a document with a coherent prepositional content. The links represent relationships between the nodes. Both the nodes and links are made accessible to clients via its Universal Resource Name (URN) which serves as a location independent identifier for the World Wide Web. A KBCW represents a set of logical statements summarizing the content of the documents as well as the relationships among the contents of the documents.

A KBCW is an evolving knowledge base: clients not only browse its contents, they add information to it. They add information by creating new nodes, creating links between existing nodes, or by creating new nodes and linking them to existing nodes. The creation of a link between nodes is, in effect, an act of discourse, relating two existing pieces of information. The KBCW also evolves as facilitator agents within the KBCW itself make inferences using background knowledge-bases, semantic representations of the contents of the nodes and the built-in semantics of the links. Different choices of link types and different background knowledge bases lead to KBCW systems tailored to different domains.

One particularly important relationship represented in all KBCW systems is the “*précis*”, a short annotation of a larger document. A *précis* is written in natural language (English in our case) with the intention of being parsed by a natural language processing system (START). *Précis* nodes are connected by “*précis*” links to the document that they annotate; a document may have many *précis* nodes attached to it. The text in a *précis* node is parsed and interpreted by the KBCW system; the resulting semantic representation is stored and made accessible for inferences by computational agents within the KBCW system.

Finally, the KBCW also includes knowledge of how an organization processes information and of how the organization is structured. The first part of this consists of organization plans, partially ordered sets of steps used to process information and make decisions. Each step of the plan has its rules of interaction. For example, in a brainstorming session, moving to close the discussion is not permitted; on the other hand, in finalizing a decision, moving to consider a new option is highly discouraged. These rules of interaction are enforced by only making available only certain types of links (and thus only certain types of discourse elements) during each phase of the project. Since elements of discourse are represented by the links made available to a client, the choice of link types amounts to a choice about the allowable forms of discourse.

Each step of an organizational plan also includes workflow plans that dictate how information is distributed.

A KBCW also includes knowledge of organizational structure, particularly the decomposition into multiple hierarchies (or DAG's) representing reporting and authority structures and detailed descriptions of each individual's interests, responsibilities and expertise. Choices about these representational elements allow the KBCW to be tailored to different organizations with diverse strategies for interaction and workflow.

In this section we focus on a KBCW system tailored to the needs of intelligence interpretation. Our focus is on the architecture of the KBCW system; we therefore illustrate our system with an intentionally "tongue in cheek" example of a possible nuclear breakout, being conducted by a rogue state, acting through intermediaries to acquire strategic information.

The Scenario

Consider an intelligence analyst who focuses on financial literature. Each day, such an analyst spends a sizable part of his day sitting at his desk reading open source literature. When he finds something interesting, or anomalous, he makes some notes about it and moves on to other articles. Often, important information is lost in the vast flood of literature that he pours through; often it fails to get correlated with information available to other analysts working from other perspectives.

In our scenario the analyst is looking at an article on trading in the precious metals market (figure 1) that discusses why there appears to be a bull market in Beryllium. The article makes reference to the "Whiplash Group" and other heavy buyers. The president of the Whiplash group is quoted as believing that the future is bright, but he doesn't explain why he believes that. This particular section of the article is only a few sentences in a much larger article. However, this section catches the analyst's attention and he decides to make a note about it. In our KBCW system, this is done by adding a new node, connected by a précis link to the section of the original article that caught his attention. He brings up the annotation web page and types the following annotation:

"Snidely Whiplash is buying unusual amounts of plutonium"

The annotation is parsed and interpreted; it's semantic representation is entered into the KBCW knowledge base. In many cases, the process would stop at this point. However, in this case, the background knowledge base of the system is capable of drawing several inferences. First of all, it knows that Beryllium is a strategic material (it is used in nuclear weapons processing). Secondly, it knows that there is particular organizational process that should be initiated any time

large quantities of a strategic material are acquired by an individual. This process consists of making a request for a dossier check on the individual; to do this, a request node is created in the KBCW and a monitor (a type of software agent) is created to wait for a response to the request. Finally, the system searches its personnel models to find a person capable of performing the dossier check (and who is available). Once the individual is identified the system must figure out the most appropriate means for actually getting the request to that person; this is an example of mapping an abstract service request into a concrete action. In this case, email is deemed to be appropriate and the request is emailed to the individual. When the response is produced the monitor will create a link in the web indicating that the response satisfies the request. At this point the contents of the KBCW are as shown in Figure 1.

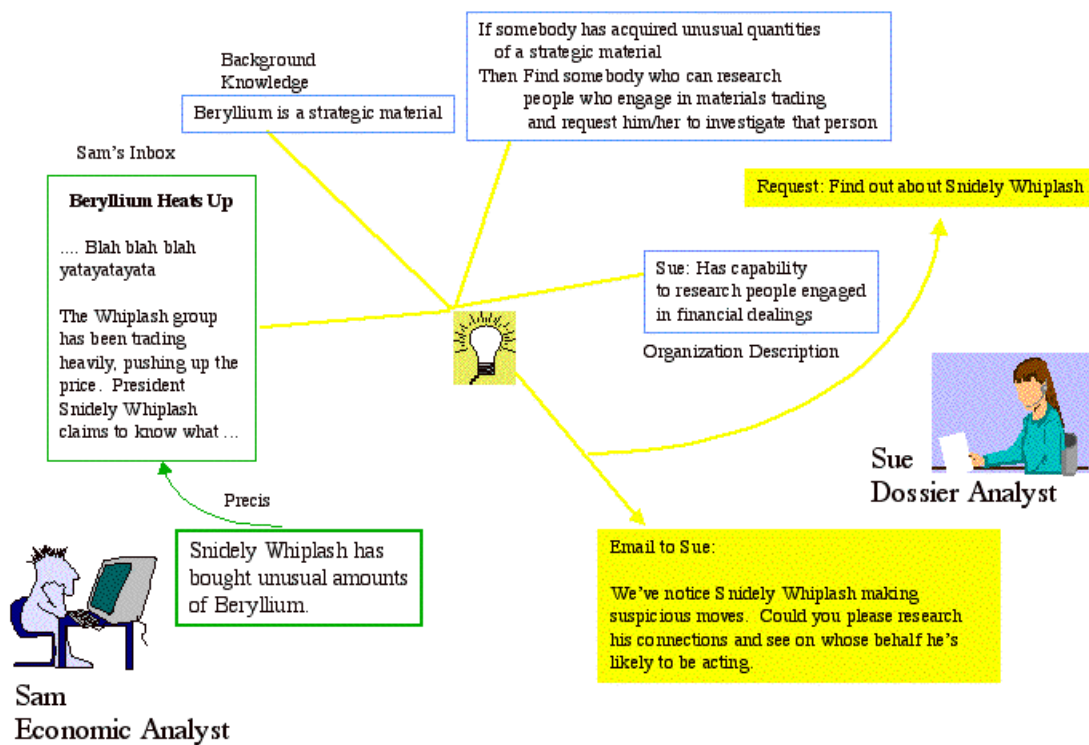


Figure 1: KBCW Contents

Sue has received the request from the system to research Snidely Whiplash. One tool she has available is the START natural language system. Earlier we saw that START is used to parse and create a semantic representation of a précis node. In that case, the semantic representation was used to facilitate inferences, in particular, the inference that dossier research was needed. In addition, the semantic representation of a précis node is used for retrieval. Natural language queries are parsed and converted into semantic representation; the representation is then matched to that of each précis in the KBCW. Each précis that matches the query has an associated document and this is retrieved in response to the query. In addition to the semantic-based representation used by START, Sue also has available full-text search capabilities. Using these, she

retrieves information about Snidely Whiplash and creates a dossier document, represented by a node in the KBCW. She annotates this node with a précis, that summarizes the information that is particularly relevant to the request.

In researching Snidely Whiplash, Sue discovers that he has acted as a front for several rogue states. Although, most of these contacts are believed to be inactive, he is believed to still be associated with Iran. Sue's annotation indicates that Snidely Whiplash is believed to be a member of Iran's procurement network. This is done by associating with the précis node an estimate of certainty that is used by Bayesian inference mechanisms within the KBCW; we will return to Bayesian inference techniques later in this section.

When Sue creates the dossier node, the monitor associated with the request for the dossier research notices it, and creates a link stating that the dossier satisfies the original request. The text in the précis is parsed and converted into semantic representation. This representation, together with rules in the background knowledge base facilitates the simple inference that Iran might have Beryllium (since Snidely Whiplash is a member of their procurement network and he has acquired unusual amounts of Beryllium).

This conclusion interacts with other information in the KBCW that has resulted from other analysts' interpretations of other documents. One such analyst has noticed in an article on the world wide nuclear power industry that Iran has acquired more fuel rods than are needed for the operation of its current commercial power plants. This analyst had created a précis stating that Iran has extra fuel rods. Yet another analyst has annotated an article on Mid-East politics with the comment that tensions are growing between Iran and its neighbor Iraq.

Further inferences are now drawn using the semantic representations in the précis nodes and rules in the background knowledge base. First, it is deduced that Iran might be capable of producing Plutonium (because given extra nuclear fuel rods and Beryllium and substantial technical expertise it is possible to produce Plutonium). Second it is inferred that since tensions are growing and Iran has the capability to produce Plutonium there is potential crisis in the Gulf area. One possible hypothesis about what might happen is that Iran might use its capability to actually produce nuclear weapons and possibly to attack Iraq.

These inferences are captured in the KBCW by creating:

1. an "issue" node that represents the possibility of a crises between Iran and Iraq;
2. an "hypothesis" node that represents the possibility of Iran producing and using nuclear weapons;

3. A link between these two nodes, that states the second node is a possible hypothesis about the issue in the first node;
4. A link saying that the reason for believing the hypothesis are the fact that Iran can make Plutonium and that tensions between Iran and Iraq are growing. At this point, the contents of the KBCW are as shown in Figure 2

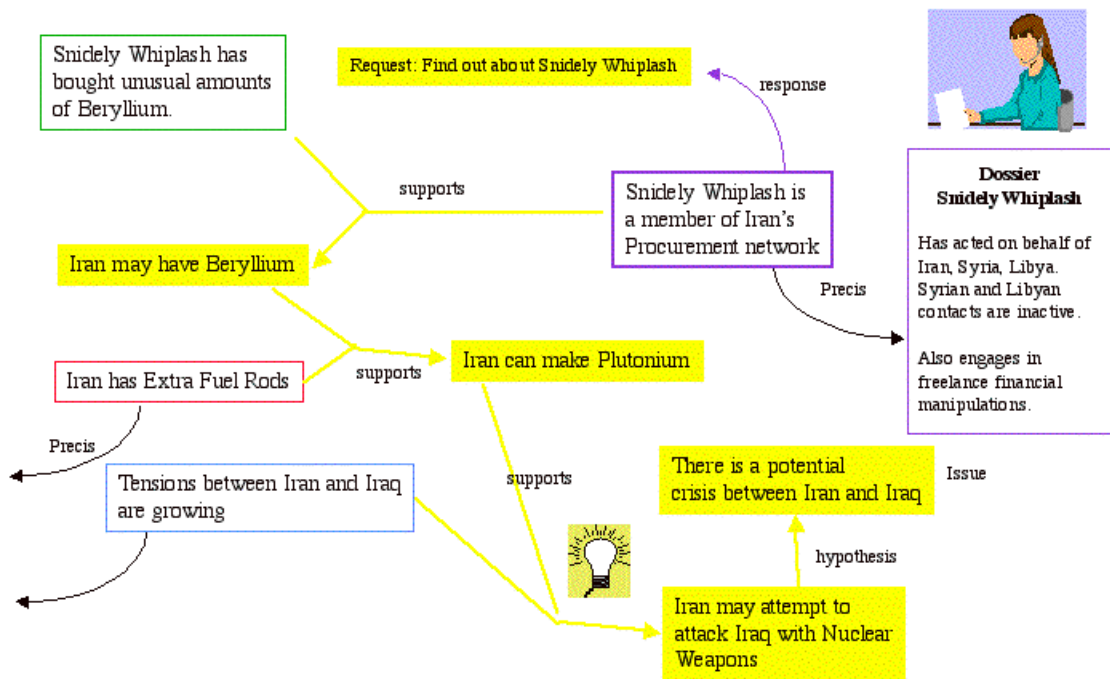


Figure 2: KBCW Contents

The system has created the issue and hypothesis nodes shown in Figure 2. However, its knowledge base also tells it that it is necessary to instantiate an organizational process whose goal is to gain a better understanding of the issue, to develop alternative hypotheses and to weigh the evidence supporting each possible hypothesis. This process has requirements for a number of participants who are chosen based on their expertise and availability. The process has several steps, arranged hierarchically. At the top level, there are two steps, the first is a brainstorming step aimed at elaborating the hypothesis set. This step is also aimed at mustering arguments for and against each position, thereby effecting the confidence that the system has in each hypothesis. The second step takes the first as input and follows it sequentially. Its goal is to plan a response to the crisis. Since the two steps have different purposes, they follow different rules as represented by the set of node and link types made available to the participants. This plan structure and the inferences that led to it are shown in Figure 3

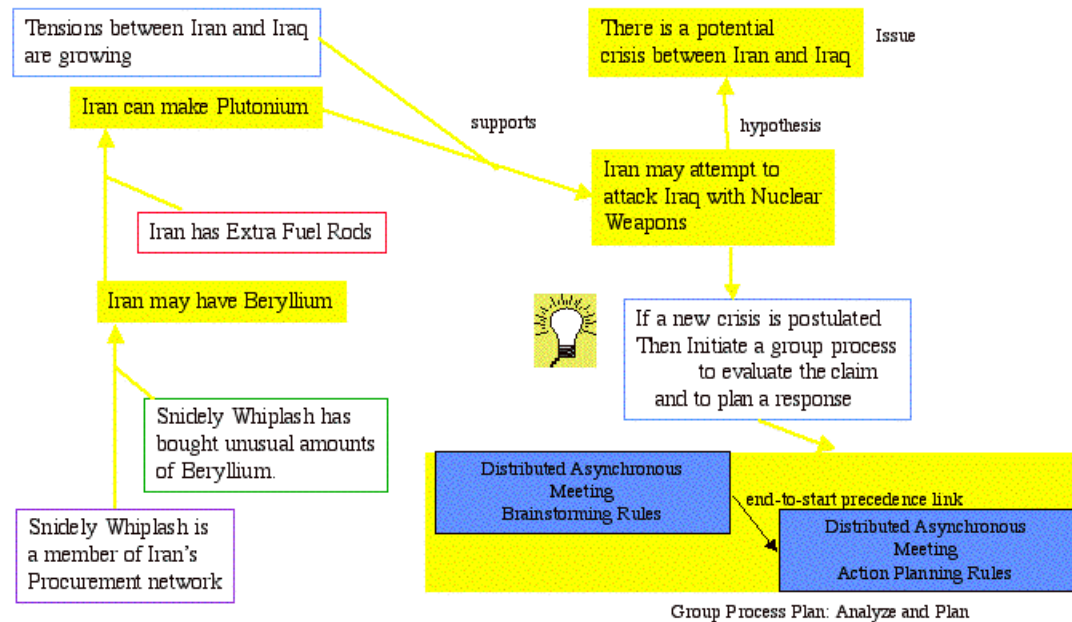


Figure 3: Plan Structure

The plan is structured hierarchically. The first step itself has two sub-steps. In the first of these participants are selected and invited to participate in the process. As before, the system must determine an appropriate technique for transmitting the invitation. Although there might be a number of different possibilities, email is again the most useful technique; each participant is sent an email message asking for their help. These email messages contain the URN of the issue that is the focus of the discussion, as is shown in Figure 4

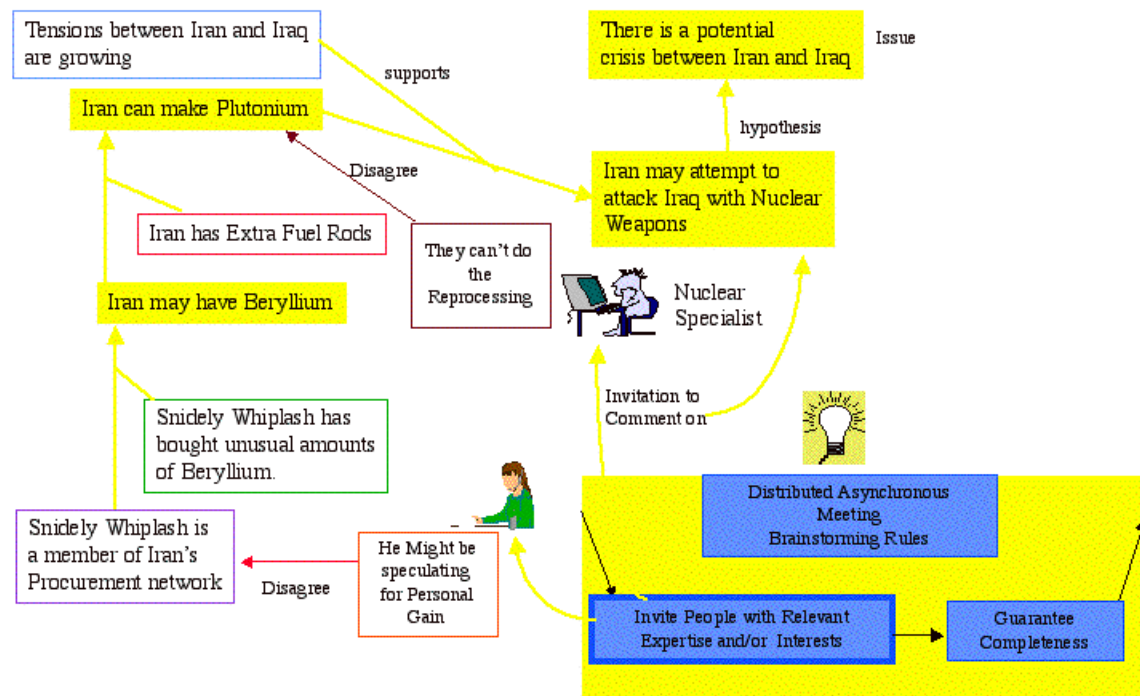


Figure 4: URN

Figure 4 also shows the results of the participation. A nuclear expert invited into the discussion makes an argument against part of the reasoning process, stating that he doesn't believe that Iran is capable of performing the technically difficult Plutonium reprocessing step required. This argument is linked to the original node by a "Disagreement" link, the opposite of a "Support" link. The KBCW system helps the nuclear, regional and political experts invited into the discussion to continue their discussion; it relays to each participant all the comments relevant to their interests. Since the system wants to make sure that all participants see these changes to the KBCW structure; it sends an email to each participant as each new node and link is entered. Browsing tools allow them to inspect the KBCW web structure around each of these areas of change. This "asynchronous meeting" has a time limit; at the end of this time, the process moves into the second sub-step, which is to guarantee completeness of discussion.

Completeness of discussion is measured in two ways. First, there are coarse measures such as the number of hypotheses attached to the issue node, the number of arguments mustered, etc. There is also a statistical measure: the entropy (or information content) of the hypothesis set. If arguments have been mustered effectively on all sides of the issue, then the information content (computed as $\text{Sum}(P * \text{Log}(P))$) should be maximized. If the KBCW process facilitator decides that inadequate discussion has taken place, it then contacts

the participants and urges them to try to explore the issue further, as is shown in Figure 5.

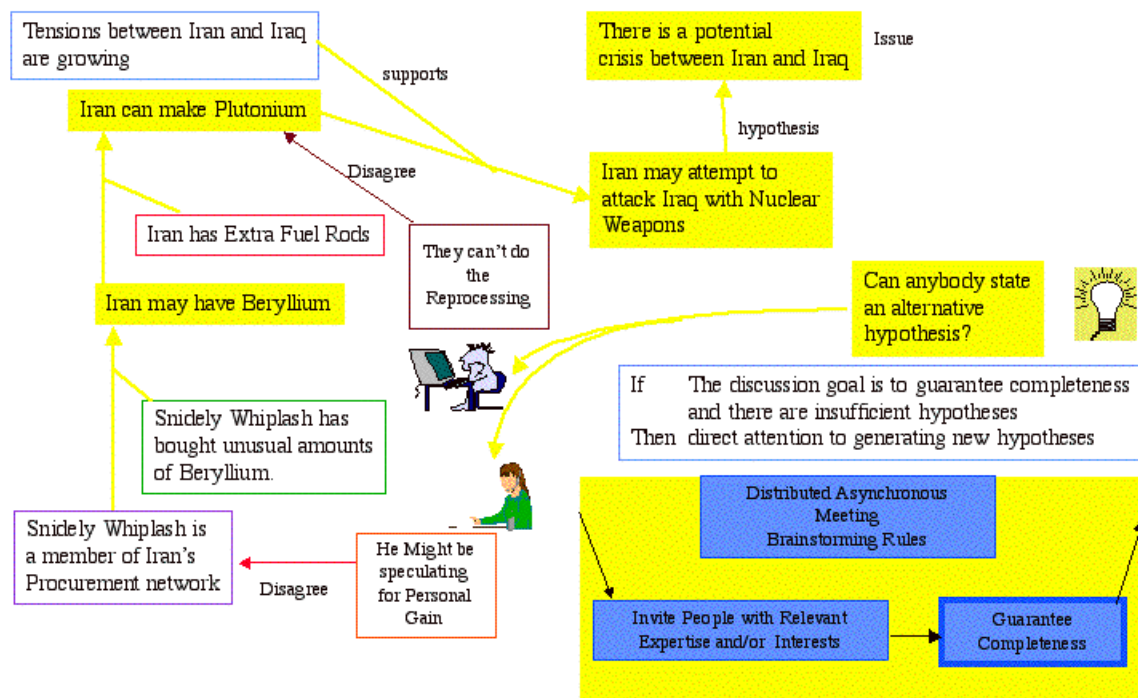


Figure 5: Inadequate Discussion

Bayesian Processing

A subset of the link types is concerned with representing issues, hypotheses and arguments in favor and against these hypotheses. These links create an evidential chain, linking observations (e.g. statements in articles, reports from direct sources) to conclusions. So far we have been referring to the propositions represented by each node as if they were binary logical propositions. However, in the world of intelligence interpretation, this would be unreasonable; everything is open to doubt, uncertainty and interpretation. Consequently, each node in the KBCW has an associated probability while each link associated with evidential reasoning (e.g. *supports*, *denies*) has an associated conditional probability. A special type of node is used to represent a logical conjunction of other nodes; otherwise when multiple links terminate at a common node, they are taken as disjunctive support.

When users create a node they associate with it an estimate of certainty. In the current system this is just the probability; however, a more intuitive representation for users would be the log-likelihood about which people seem to have better intuitions. Similarly, all inferences made by agents within the KBCW

system are represented by nodes with probabilities that are connected by links with conditional probabilities. This structure is isomorphic to a Bayesian inference network. The KBCW system therefore extracts from the overall KBCW representation the subset of nodes and links that participate in Bayesian reasoning and creates a Bayesian network in the IDEAL system (using its implementation of the Jensen algorithm). This allows it to calculate the probabilities of the all the nodes in the evidential chain, ultimately concluding with the individual hypotheses associated with the issue under discussion. This Bayesian network is shown in Figure 6. At the top of this graphical presentation of the Bayesian network is the issue with the various hypotheses immediately underneath it. Each hypothesis has a posterior probability shown with it. Each hypothesis is in turn supported by evidence, in this case by the proposition that Iran is building nuclear weapons. The support link to each hypothesis has a different conditional probability attached to it, accounting for the distinct posterior probabilities of each hypothesis. By moving down the evidence chain, we see how and where the information entered by the different players during the scenario enters into the evidential reasoning process.

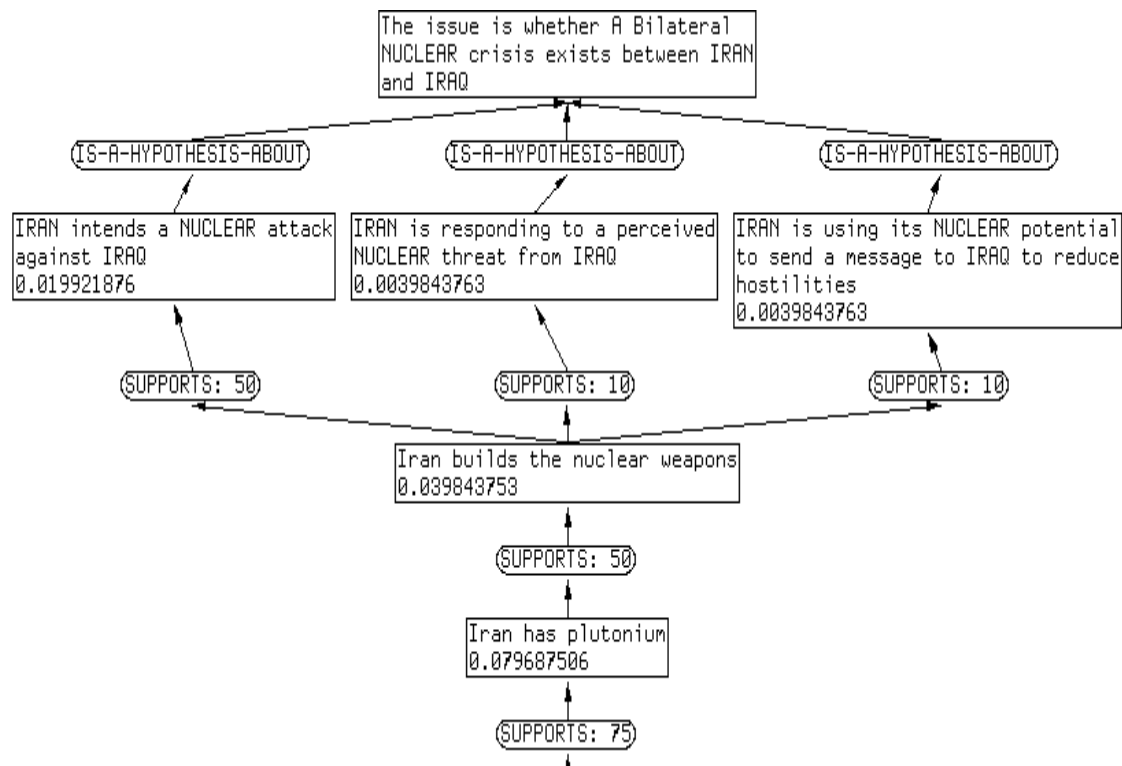


Figure 6: Bayesian Network

In Figure 7, are two conjunctive nodes labeled "Support of Iran has plutonium" and "Support of Iran has Snidely Whiplash's beryllium". Each of these has the force of a "logical and"; it can be seen that the probability of each of these nodes is just the product of the probabilities of the supporting nodes.

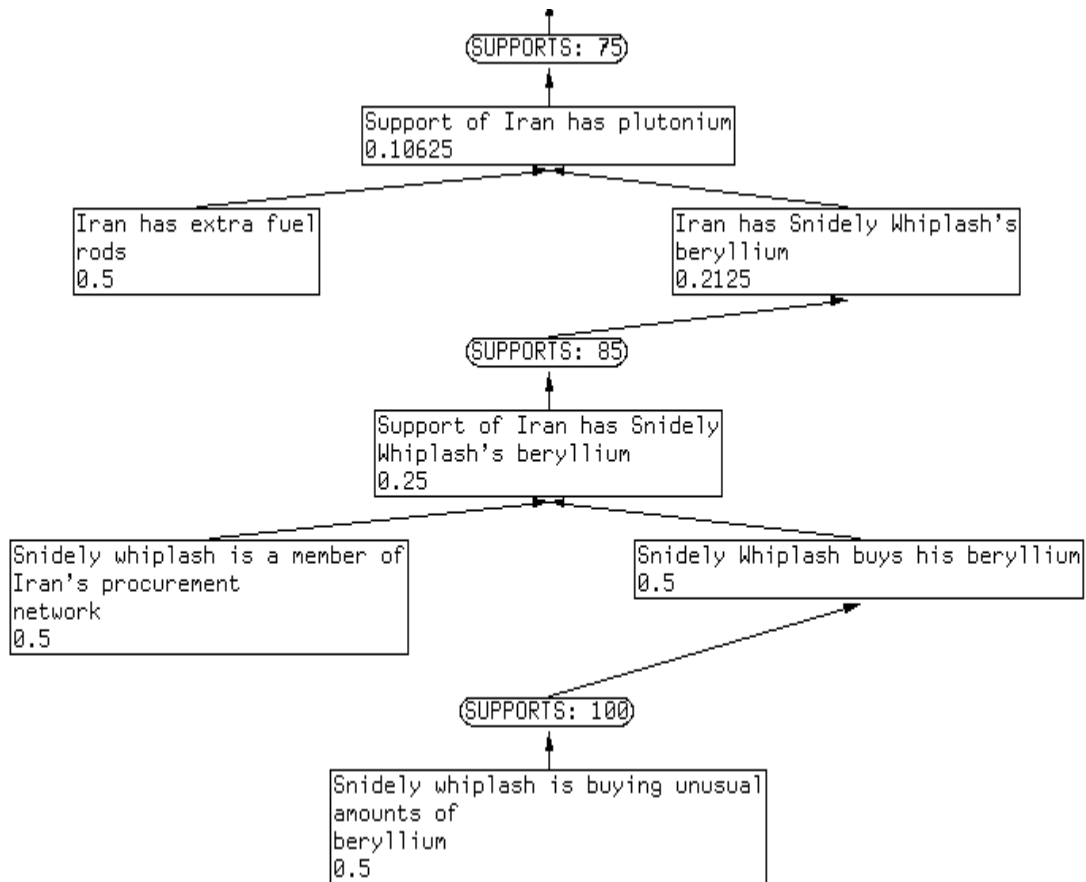


Figure 7: 'Logical and' Bayesian Network

The Planning Phase

The next stage of the decision making process involves action planning. In this phase the planners consider what courses of action are likely to lead to the best results. However, this planning takes place in the uncertain context of the first (analysis) phase. At this stage each hypothesis has a posterior probability representing how likely it is given the consensus estimates developed during the analysis phase. Each course of action has a range of outcomes and each of these outcomes has a value. However the likelihood of each outcome is conditionally dependent on each of the hypotheses. This dependence is naturally captured in an extended form of Bayesian network that calculates the expected value of each decision node (i.e. each course of action).

The KBCW framework can also support this stage of process, although in this case a different set of node and link types is used to express courses of action, outcomes, the values of outcomes and the conditional dependence of outcomes on the hypothesis.

The entire KBCW network in effect constitutes a briefing book that can be passed on to the ultimate decision-makers. The network represents the full chain leading from evidence to recommended course of action. However, no decision-maker would take this recommendation at face value. Instead, he or she would want to look at the evidence mustered, the probability values assigned and the structure of the decision space. He might want to change some of these values, or exclude the input of certain of the contributors who represent one or another coherent viewpoint. All of these are simple extensions of the current KBCW structure.

Technologies used in the scenario

Throughout the scenario we made reference to a set of core technologies that collectively lead to the power of the Knowledge-based collaboration web architecture. These include:

1. The Comlink infrastructure for document management, indexing and distribution. This provides stable storage for the documents managed by the KBCW system, a taxonomic system for categorizing the documents, automatic (statistical) tools for tagging documents with their appropriate taxonomic labels and tools for automatically generating HTML for viewing the documents through the World-Wide Web.
2. URN and link based assertion infrastructure.
3. START natural language understanding system. This is used for information retrieval uses natural language queries as well as for parsing the text in Précis nodes.
4. Service Mapping and Resource Management. This transforms requests for abstract services into concrete plans using specific resources and then evaluates each possible plan to determine which renders the optimum tradeoff between cost of the resources and benefit delivered to the user.

5. Project management tools for describing organizational plans, resource requirements, loading and commitment levels of resources (and people), capabilities of resources, interests and responsibilities of individuals. These are shown in Figure 8 through Figure 10.

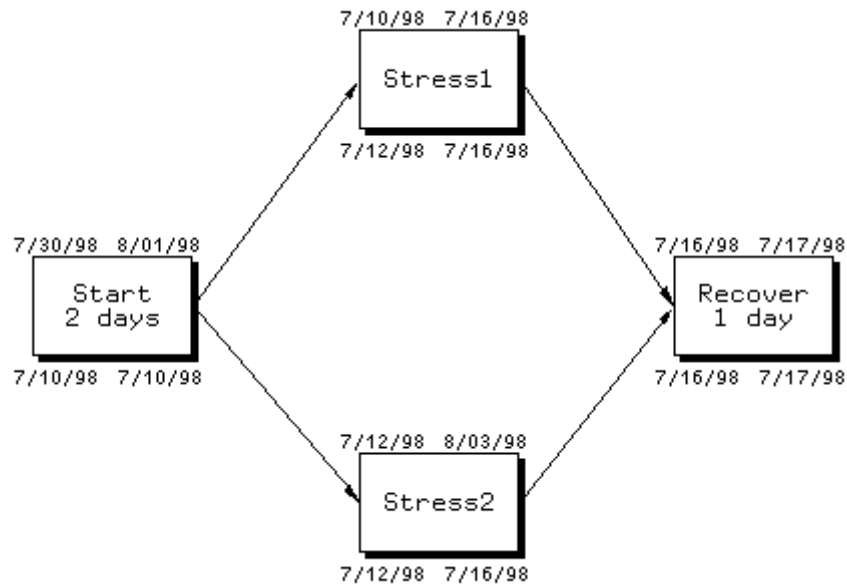


Figure 8: Project Management Tools

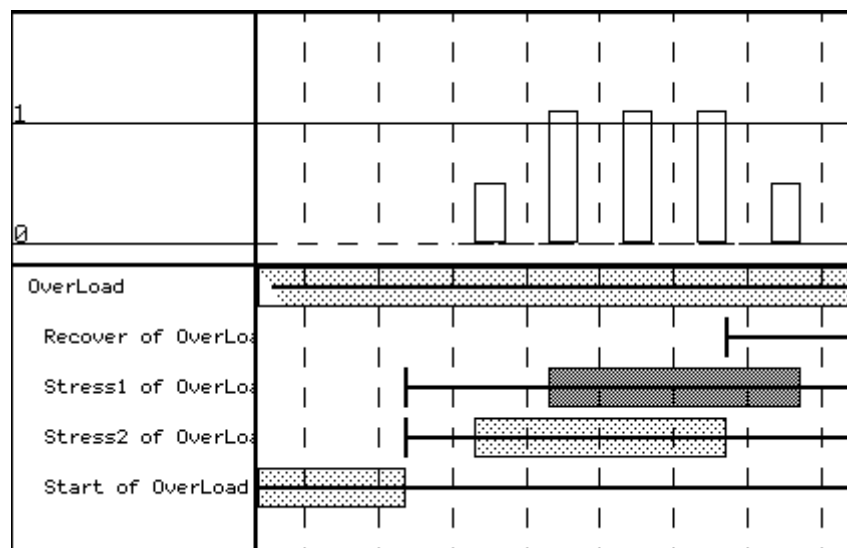


Figure 9: Project Management Tools

Project Planner

Employee Editor

ID Number	Name	Calendar	SwapCost	Capabilities	Department	Manager
boris@...	Boris Katz	boris	0.3	PRS	AI Lab	brooks...
brooks...	Rodney Brooks	Default	0.0	Full Prof	AI Lab	
cvince...	Chris Vincent	Default	0.0	Grad	AI Lab	hes@ai...
hes@ai...	Howard Shrobe	Default	0.1	PRS	AI Lab	brooks...
jcn@ai...	John Mallery	Default	0.0	RS	AI Lab	hes@ai...
rhhu@ai...	Roger Hurwitz	Default	0.5	RS	AI Lab	hes@ai...
rsd@ai...	Rodney Daughtrey	Default	0.7	RS	AI Lab	hes@ai...
sfeish...	Sue Felshin	Default	0.1	RS	AI Lab	boris@...

☒

CreateDelete

Capability Codes

Assoc Prof, Asst Prof, Full Prof, Grad, PI, PostDoc, Prof, PRS, RS, SRS

☒

☐

Figure 10: Project Management Tools

Section 4: Transition to Project Oxygen

The Intelligent Room is a multi-modal facility for complex user interactions and some of our activities have been concerned with integrating our technology with the Intelligent Room. During the course of our project, the Intelligent Room became an important component of the newly formed MIT Project Oxygen. MIT Project Oxygen is a consortium of six companies¹, the MIT AI Lab and MIT's Lab for Computer Science. DARPA ITO sponsored Project Oxygen as part of its Ubiquitous Computing effort. Project Oxygen is concerned with *human-centered, ubiquitous computing*. It is built around three main technologies: 1) Personal computing devices (the H-21 or handheld device for the high-end personal computing devices from HP and Compaq are the prototypes); 2) Environmentally embedded computing (the E-21 for which the Intelligent Room is the prototype); 3) An advanced adaptive networking infrastructure linking these together (the N-21). These devices are utilized by distributed computing applications that are built on a goal-oriented computing framework derived in part from work on *Service Mapping* in our project (more about this below).

One key technology focus area for Project Oxygen is that of collaboration. Our KBCW framework was chosen as a basic framework to build on. The World-Wide Web Consortium (W3C) had begun standardization of technologies that are now called "The Semantic Web" that are based in part on the representations we use in START, RELATUS (our two Natural Language Processing Systems), as well as the KBCW web structure. It was therefore natural for us to work with the W3C within Project Oxygen on the further development of collaboration webs.

Our work in the KBCW project had begun with a focus on use of Web technology, standard browsers and asynchronous collaborations (i.e. people working together although separated in time and place). Although we feel that basic concepts developed in this effort are correct, it was also noticeable that interfaces based on standard Web Browser technologies are not particularly natural or fluid. Project Oxygen offered us much more natural interfaces; in particular the Intelligent Room allows us to explore the user of Speech, Sketching and Machine Vision as input modalities for both synchronous and asynchronous collaborative interactions.

In the rest of this chapter we will first describe a collaboration system, called The Meeting Manager, developed for the Intelligent Room that is based on the KBCW architecture for collaboration. We will then talk about the Service Mapping framework developed by Krzysztof Gajos in the KBCW project that is being used as one of the key components of an overall software architecture for Project Oxygen.

¹ The companies are NTT, Phillips, Nokia, Hewlett-Packard, Acer, & Delta Electronics. Compaq has very close association but is not a sponsor.

The Meeting Manager

Although the Intelligence interpretation scenario we presented earlier involves the use of KBCW technology for asynchronous collaboration, this is not the only format in which team-based collaboration takes place. Indeed, synchronous interactions, in the form of meetings and pair-wise group discussions are a key component of the collaborative process.

We chose to focus our attention on the use of the multi-modal interaction capabilities of the Intelligent room; a natural focus for these technologies is the design review meeting. Design reviews are typically concerned with exploring *design issues*. Each such issue may have associated with it a number of *positions*, and supporting or opposing each position are *arguments*. These basic concepts are linked together in a KBCW web just as were the issues, hypotheses, and arguments of the intelligence interpretation system. Each meeting has an agenda and each agenda item has both a time budget and a topic. Associated with each topic are a number of issues to visit. Finally, during such meetings people make *commitments* to undertake certain activities such as to explore the evidence for a particular position. Each commitment is associated with the individual who makes the commitment, with related issues, and with agenda items during which it was discussed. All of these concepts and relationships are represented in a KBCW structure.

The Meeting Manager system that we built utilizes most of the technologies available in the Intelligent Room. Figure 11 shows the system in operation. There are four people involved in the meeting, projectors are used to create displays on two walls, speech input is the primary means of interaction, and a sketch understanding system is used to capture design sketches, software architecture diagrams, etc. Each meeting is captured as a quicktime movie (using the cameras and microphones in the room). Nodes in the KBCW web are associated with fragments of this movie transcript (e.g. a commitment is associated with the fragment of the quicktime movie during which the commitment was made). Issues, Positions and Arguments are often discussed in several successive meetings, so the KBCW nodes for each refer to many different meeting transcripts, each captured as a fragment of a Quicktime™ movie.

Figure 11 shows an example of how the Meeting Manager makes use of this structure. In the foreground, the participants are reviewing the commitments from a prior meeting. One of the participants missed the meeting being discussed and wanted an update. The easiest way to do that was to replay the section of the movie transcript of the previous meeting that is relevant to the commitment being reviewed. This is being shown on the screen behind the participants.

The moderator is wearing a headset microphone and issues voice commands to the Meeting Manager system. His first command is to review the commitments from the prior meeting, thereby establishing the context. He can then ask to show the movie fragment associated with that context, which in this case shows the part of the prior meeting when the commitment was made.



Figure 11: Participants view a commitment being made in a prior meeting

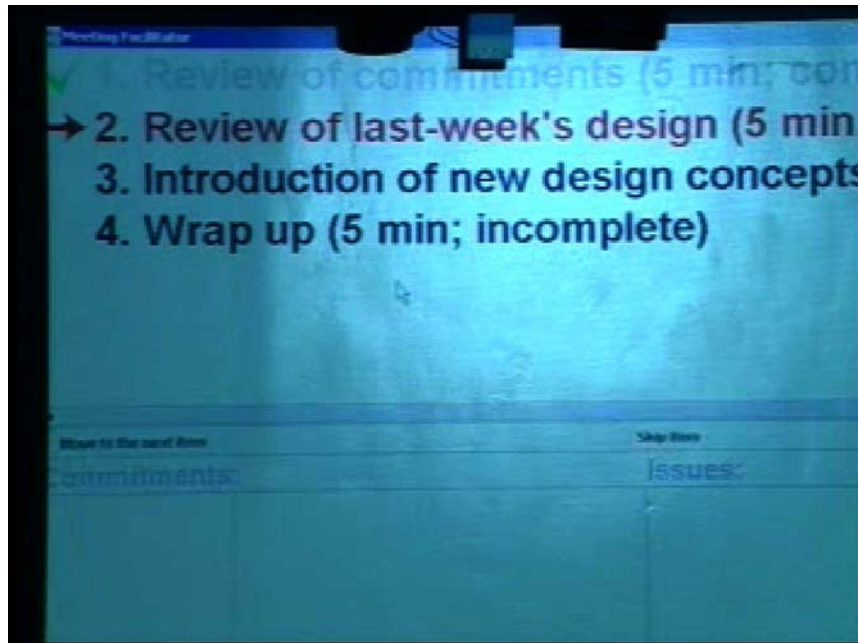


Figure 12: A view of the current meetings Agenda

Figure 12 show a view of one of the Meeting Manager's displays, in the case the agenda for the current meeting. Other displays show commitments and the issues, positions and arguments. A final display is a viewer for browsing commitments, issues, etc. from other meetings.

A Meeting Web

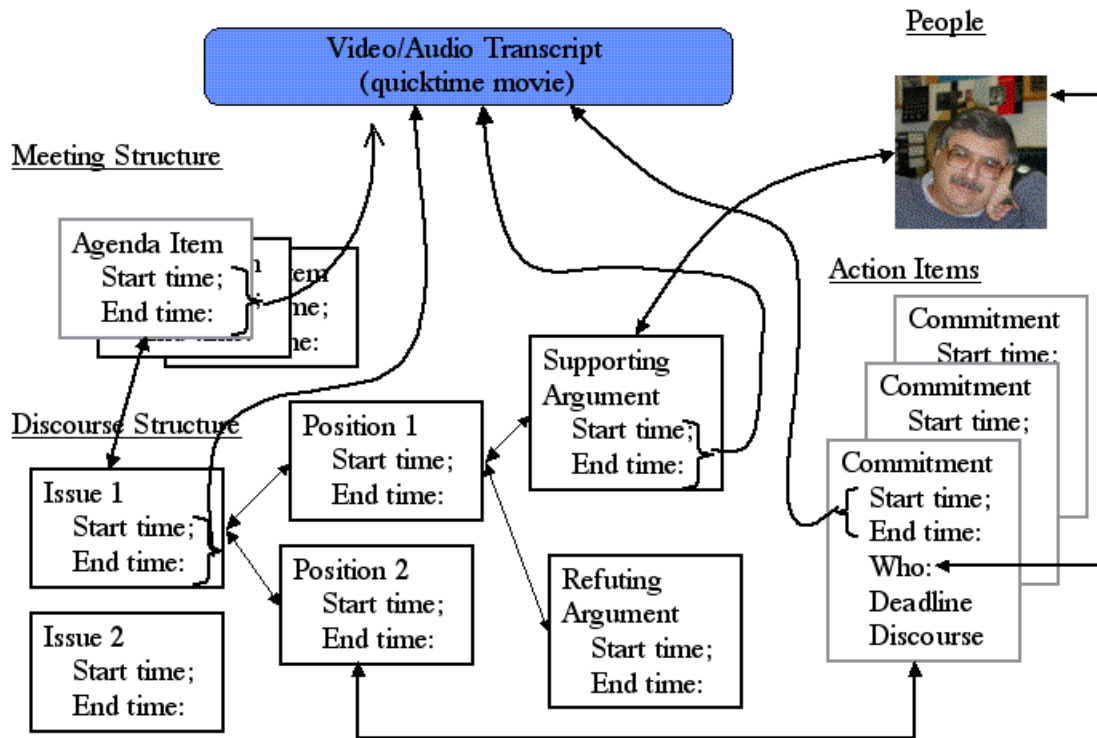


Figure 13: The KBCW web structure used by the Meeting Manager

Each element of this web structure contains “movie fragment references”, i.e. references to the parts of the Quicktime™ movie in which the particular element was discussed. Each fragment contains a start and stop time as well as the URN of the particular Quicktime™ movie. Start and stop times are determined by the occurrence of “significant events.” Significant events include changing an agenda item, changing the focus in the issue structure, and the making of a commitment. In addition, the nodes representing commitments also include reference to the active focus of the issue structure when the commitment was made and the agenda item that was current when the commitment was made. Agenda items contain references to issue structure elements that were visited during the course of the discussion of that agenda item. All the references are symmetric (i.e. for every reference there is a corresponding backward reference). Finally there are KBCW nodes for the people involved which are also referred to by the other nodes. Issue structure nodes refer to people who spoke to the issue (position or argument) and commitment nodes refer to the person who made the commitment (and to the person to whom the commitment was made, if relevant). As in the Intelligence Interpretation system, the KBCW nodes that describe individuals contains interests, expertise, responsibilities and roles within the organization.

Figure 14 shows the displays normally presented by the meeting manager. The lower left display is the issue structure. From left to right it shows issues, positions taken on those issues, and arguments for and against each position. Notice that one element of the graphical display is in bold. This is the **current focus**. The current focus is changed during the discussion to reflect the actual focus of the discourse. This can be done either by gesture (using a laser pointer at the moment, and figure gestures once the machine vision system can provide this capability). As mentioned above the current focus is stored in commitment nodes as an aid in trying to understand the commitment during later browsing.

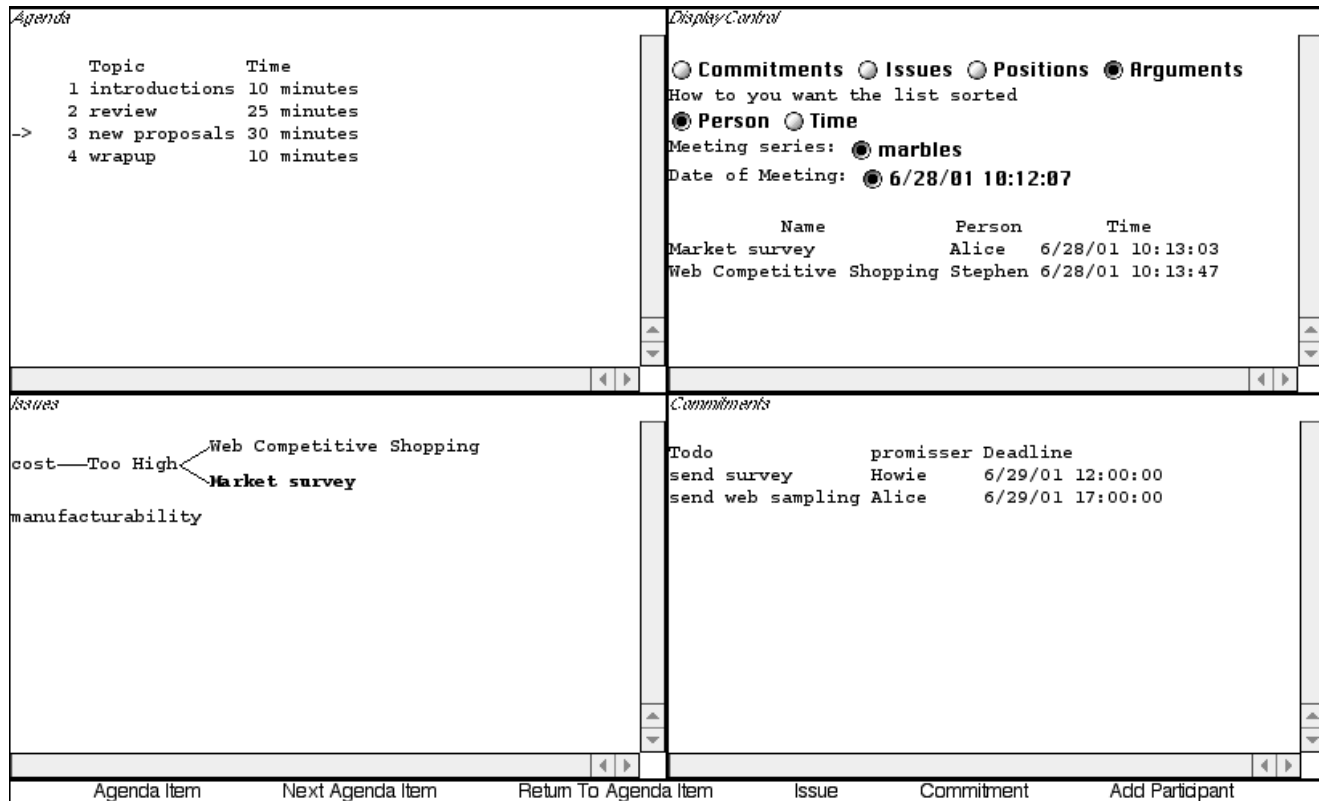


Figure 14: The displays maintained by the Meeting Manager

The upper right display is the viewer for browsing through prior meetings. It allows these to be retrieved and sorted either by which people are associated with the item or by the time at which the item was recorded. The lower right display shows commitments made during the current meeting, the upper left is the agenda. The meeting leader can navigate through any of these structures using either voice commands or by pointing with a normal laser pointer (this is tracked by a machine vision system; in the future we hope to use machine vision to track hand gestures as well).

We contrasted the synchronous nature of the collaboration activities supported by the Meeting Manager and the asynchronous activities supported in our

Intelligence Interpretation system. However this is an not exactly correct. The Meeting Manager creates a KBCW structure during the synchronous activity (i.e. the meeting) but this structure may be browsed and annotated in an asynchronous manner using the existing capabilities.

During future effort in Project Oxygen, we hope to make these offline browsers use more of the multi-modal capabilities provided during the meeting itself. Since we now have many of our offices outfitted with capabilities similar to (but more modest than) those in our multi-modal meeting facility, this should be possible in the near term.

Sketch Understanding

One particularly interesting capability provided in our multi-modal, Intelligent Room facility is a system for understanding sketches. We use Mimio devices (from Virtual Ink) to capture the strokes of a marker on the whiteboard (in a group meeting) or tablets to capture pen strokes (when working individually). Low level processing recognizes basic geometric objects as they are drawn (e.g. lines, circles) while higher level processing aggregates these into semantically meaningful elements for the domain of application. Currently, we have a fully functional system that interprets simple mechanical drawings.

Figure 15 shows the system being used during the meeting to sketch a “marbles game” (which is the design problem being worked on in the meeting). As the user draws basic shapes these are recognized and cleaned up by the drawing system. For example, the user’s drawing elements may not be straight lines or perfect circles, but the projected image is cleaned up as the elements are drawn. (In this mode, a “null marker”, one that doesn’t actually write is used, the computer interprets the strokes and the projector projects what the user meant to draw).

More significantly, the system also interprets the strokes as semantically meaningful elements: the “squiggles” are springs, the line and touching circle is a pendulum etc. Elements with an X on them are attached to the fixed frame.

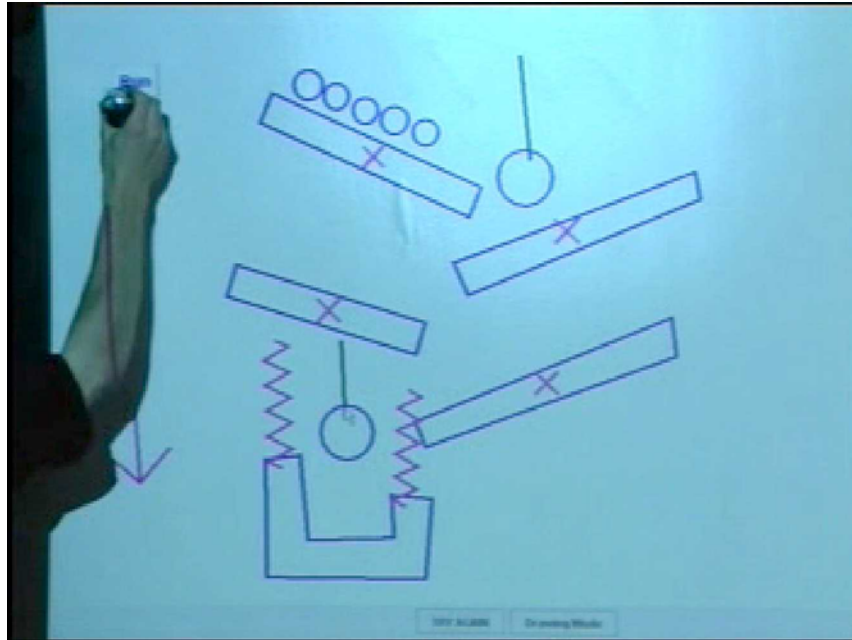


Figure 15: The Drawing System Captures and Cleans Strokes

In the case of mechanical drawings the semantic understanding can be dramatically illustrated by feeding the interpretation of the drawing into a mechanical simulator (in this case a commercial system called Working Model). Figure 16 shows a simulation of the drawing shown in Figure 15.

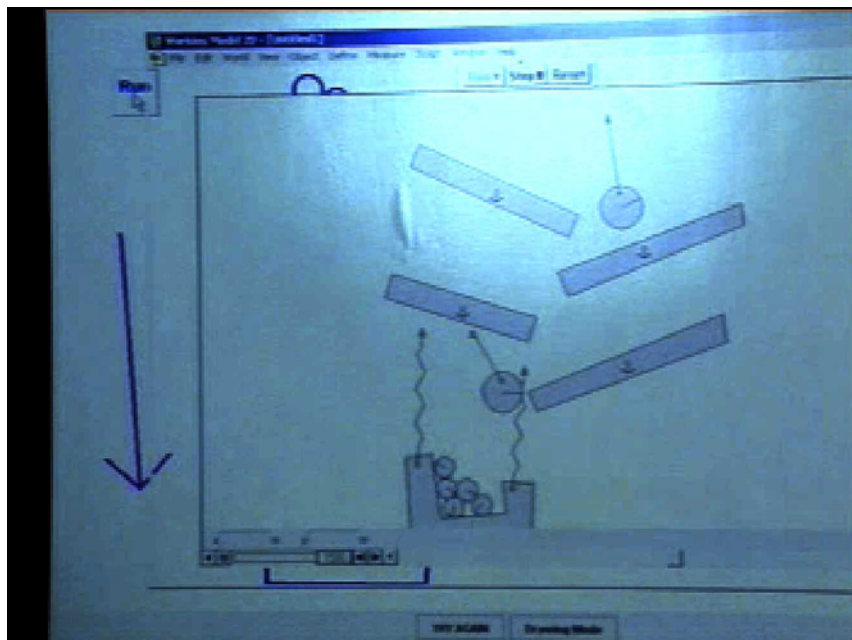


Figure 16: The Sketch Understanding System Creates a Simulation

We are currently working on other sketch interpretation systems for other domain, including software design diagrams (e.g. UML) organizational diagrams, architectural floor plans and military course of action diagrams.

Service Mapping and Resource Management

A current theme in all the uses of KBCW technology has been the need to systematically manage resources. Although collaborations involve groups of people working in a team with a common goal, they are doing different tasks at the same time and these tasks may compete for common resources. When this happens it is important that the allocation of resource allows the team's goals to be achieved efficiently, even if this inconveniences some member of the team. Indeed, if there is an actual conflict over a resource, then some member of the team performing a less important task will have to incur some inconvenience in order to allow another member who is performing a more important task to function at a higher level. We visited this issue in two steps, the first was George Dolina's Master's thesis which considered the issue in the abstract and the second was Krzysztof Gajos' Master of Engineering thesis which considered this in the context of the intelligent room.

There is a second motivation behind these projects besides maximizing the efficiency of resource use. Generally speaking, an application that specifies its needed resources in very concrete terms will need to be rewritten to run in a similar environment that differs in small details. Indeed, the original software of the first Intelligent Room specified its devices quite specifically (e.g. left projector, right projector). However, when we built a second Intelligent Room with six projectors (making the names left and right projector meaningless), all the existing software had to be modified. Moreover, if a specific resource fails (e.g. a project bulb burns out), the application as written will fail to operate even though adequate resources may exist.

The solution that we developed for this was to concentrate on *services* rather than *resources*. After all, the application uses the resources to render some service (e.g. to display information) and there may be many other ways to render that same service. Thus, it is better to postpone the decision about which resources to use until run-time and instead to write applications which request abstract services. Moreover, services can be organized into a taxonomic structure and an application should request the most generic class of the service that is still consistent with its current purpose. For example, it should request information *delivery* not information display, if alternatives to displaying the information would be acceptable (e.g. speaking it out loud using a voice synthesizer).

This leads to a multi-step process that is illustrated in Figure 17. First, the application makes a request for an abstract service. Second, the system consults its service rendering library (i.e. a library of known methods for realizing services) and finds all methods that are capable of rendering the requested service. Each method is then examined in turn. Each method contains a set of

resource descriptions specifying constraints on the set of resources used to implement the method. Each method also determines the value of the parameters of the service (e.g. if the service is information delivery, then using a method that prints the information will set the “speed” parameter to slow). The resource descriptions are passed to resource pool managers for the relevant types of resource; the pool managers then return a set of resources consistent with the constraints. At this point, the system estimates a cost for these resources. It also estimates the benefit to the user rendered by using this particular method with this particular set of resources. This benefit is calculated by using a utility function provided with the service request; the inputs to this function are the values of the parameters of the service description, (which have been set by the choice of the method and resources). The final step is to compare the benefit delivered to the user to the cost of the resources consumed. That method and choice of resources which maximizes the benefit to cost ratio is the best choice.

What if a highly desirable resource is already in use for some other purpose? There are then two options. The first is to preempt the other user and allocate the resource to the newer request. To first order, this would be justified if the benefit to the newer request exceeded the benefit to the old user. However, the situation is more complex. First we must consider how important each request is to the project as a whole and weight the benefits of each potential user by their relative importance. Second, we must consider the costs incurred by the preempted user. These include the difference in quality of service rendered with the old resource versus that rendered by the best available replacement resource. An additional cost is the disruption caused by the preemption. If these two costs are exceeded by the additional benefit rendered to the new request (weighted by relative importance) then it is rational to preempt the resource and give it to the newer requestor. The other available alternative is to let the newer requestor use a less desirable resource; this would be the right decision if the additional benefit rendered to him (weighted by his importance) does not dominate the harm caused to the current user of the resource.

In effect, this process reduces service rendering to a decision-theoretic choice. Doing this maximizes flexibility, evolvability and robustness. Project Oxygen has adopted this model as part of its overall software strategy because it contributes to the adaptivity that we regard as a critical component of human centered computing.

Service Mapping is Decision Theoretic Resource Management

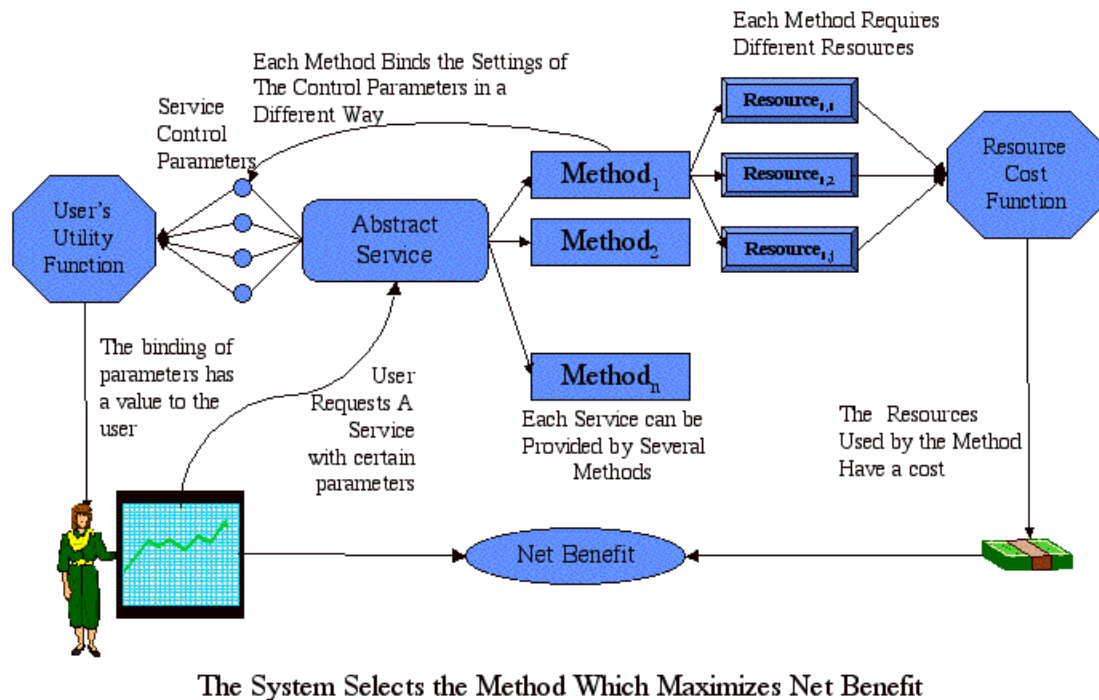


Figure 17: Services are mapped to plans. Plans are evaluated by their cost-benefit.

The service-mapping framework is also appealing because it allows other issues to be brought into consideration within the same decision making framework. Consider for example the issue raised by privacy and security policies. Generally speaking, privacy and security are in conflict with convenience and ease of use. In addition, all such policies tend to have exceptions (e.g. I don't want my location distributed freely, however if a member of my family were sick or in danger, I would my location to be accessible).

One way of dealing with this need for flexibility in privacy and security policies is to include them as part of the service-mapping framework. A method for rendering a service request that uses resources in ways that violate policies will incur a "negative benefit" during the cost-benefit analysis, making that method very unappealing, unless it also provides some positive benefit that outweighs the cost of the breach of policy. This has exactly the intended behavior: violations of policy are unlikely to occur except in some unforeseen circumstance where overall benefit is increased by breaking the policy.

Section 5: The Importance of URN

What's Wrong With Uniform Resource Locators (URLs)?

With the advent of the World Wide Web, the Hypertext transfer protocol (HTTP) and the associated resource identifiers known as URLs, or Uniform Resource Locators, have become familiar to most Americans. URLs were a critical enabler of the Web because they masked idiosyncratic syntaxes used by operating systems to access computer files with a uniform generic syntax for identifying networked resources. Its achievements notwithstanding, the design of URLs has some shortcomings that limit their range of application. In this discussion, the critique of URLs refers primarily to the widely deployed HTTP URL scheme but may also apply to other URLs with analogous syntax and semantics.

Transport Protocol Specificity

URLs mostly encode the transport protocol over which they are resolved in their scheme name (e.g., HTTP, NEWS, WAIS). Tying a resource to a protocol is a means to provide hints on how to obtain the resource in lieu of a separate mechanism for resolving identifiers. But, there is no reason to presume that a particular resource can only be available via a single transport protocol or that it may not be accessible in the future via some new transport protocol. This, then, is a design error that conflates resource identification with resource transport.

Location Specificity

URLs encode a physical location from which the resource may be obtained. Like the transport issue, there is no *a priori* reason to suppose that resource can only be available from a specific host in a specific directory and file. This commitment followed from the origins of URLs and HTTP as a uniform front end for file systems of differing operating systems. Although a useful early simplification, this conflation of physical storage location with identification presents problems when the resource moves to new locations. The problems include inability to determine document equality from identifiers alone and indefinite backward compatibility (requiring installation of HTTP forwarding redirects).

Mobility Limitations

Since identifiers are tied to specific protocols and specific physical locations on the network, problems arise when attempting to digitally sign documents. If there is a need to update the identifiers inside a document to match their current location, then the digital signatures break. For this reason, people are forced to use relative URLs in such documents, where a relative URL refers to the directory and file components minus the scheme and host/port components. But,

this solution is only partially effective and requires all the documents referred to in a digitally signed document to be collocated to the same host. That way, when the current transport protocol, local host and port, and possibly directory are merged against the relative URL, the resulting full-specified identifier will denote an accessible resource. Although this work-around can succeed for HTTP URLs, it is unclear what happens when multiple types of URLs exist in a document, for example HTTP and FTP, as merging rules only handle a single scheme. The mobility limitations follow from the fact that URLs are tied to physical location and access scheme and fail to provide an ability to determine resource equality based on identifiers alone. More generally, these defects reflect modularity problems in the overall design of URLs.

Version Omission

URLs provide no organized mechanism for referring to different versions of a resource. Again this shortcoming follows from their origin as front ends for various file systems, many of which do not maintain file versions. The binding of a URL is whatever it currently accesses at a physical location over a specific protocol, provided some resource is actually found at the location. Some heuristic methods are available to infer resource versions, based for example on modification date, but these provide only partial solutions. It is a failing of URLs that they do not allow version comparisons based exclusively on the identifiers without reference to other information such as dates, which must be obtained by accessing resource metadata over a specific protocol at a specific physical location. Any versioning schemes are left to users to devise, and consequently, one can rely on no interoperable versioning. While many applications may not need versions (especially, manual human-assisted uses like Web browsing), there are a considerable number of more sophisticated and often automatic uses that are impractical without versioned identifiers, most notably source code management and references to fragment of resources.

Fragment References

URLs make no credible provision for denoting fragments of the resources to which they refer, and indeed exhibit great confusion over the meaning of a fragment reference. The # delimiter in HTTP URLs provides a means for a browser to jump to a position in an HTML document, but it neither provides a means to generically refer to pieces of multimedia resources nor does it guarantee stability of reference across versions of a resource.

In sum URLs are an important innovation that have made possible a wide variety of human-assisted applications on the World Wide Web, but the syntax and semantics of

URLs is unsuitable for a number of interesting applications that involve long-lived stable resources that can be accessed over multiple protocols or involve fragment references and digital signatures.

How Uniform Resource Names (URN) Address the Shortcomings Of URLs

Uniform Resource Names (URNs) are late-binding identifiers that make minimal commitments to the internal structure of identifiers and completely decouple the transport protocol and physical location from identifier syntax by providing protocols for resolving identifiers to resources. Within the URN framework, specific identifier schemes are known as namespaces. The general URN specifications require a URN namespace to preface its identifier with a unique registered component that denotes the namespace and to utilize forward slash as the delimiter for hierarchical components (analogous to directory components). Beyond these requirements, a URN namespace designer is free to structure his identifiers in any way and, where relevant, to stipulate any specific semantics associated with identifier components. This generality opens many possibilities for specialized identifiers targeted at particular domains as it eliminates a number of the shortcomings of URLs.

- ❑ The availability of a resolution protocol for accessing a resource, given an identifier, eliminates from all URNs the problems of transport protocol specificity, physical location specificity as well as mobility limitations. The URN resolver can respond to a query by returning the actual resource or a URL for where it can be obtained. This late binding of the identifier to the actual resource is the key property possessed by all URNs.
- ❑ Because no hints as to the protocol or physical location of a resource need to be encoded by URNs, there is no mobility limitation. Identifiers embedded within resources do not need to change because any differential access or relocation of the data is handled by the URN resolver rather than by mutable information encoded in identifiers.
- ❑ Although URNs do not make any general provision for versioning or fragment references, the ability to define a namespace with specific syntactic and semantic properties associated with its identifiers allows URN namespace designers to handle the issues. For example, the Persistent Document Identifier (PDI) namespace developed for use with White House Electronic Publications provides both resource versioning and an extensible fragment syntax in a URN identifier intended for general-purpose use.

In sum, late binding via resolvers for identifiers and the ability to create specialized namespaces with key properties allows URNs to provide capabilities that are practically beyond the reach of URLs. Recently, URN resolution protocols have been "generalized" to include Uniform Resource Indicators (URIs), which are super class of both URLs and URNs, with a view towards providing the late binding property for URLs, especially HTTP URLs. However, the structure of HTTP URLs creates a flat namespace on the one hand

(everything belongs to the http scheme) and fails to provide chronological delegation that would allow URLs issued with the same host domain names to be resolved by different authorities based on their time of issue. Furthermore, HTTP URLs do not carry critical syntactic and semantic properties that would allow versioning and generic fragment references to work. Again, we see an example of attempting to retrofit desirable properties onto URLs, but these efforts can only be made to work within a overly restricted range of application.

New Opportunities Opened by URNs

Long-Lived Identifiers

The original idea for introducing URNs was to provide identifiers that could live longer than particular Internet transport protocols. The requirement was that the physical storage location of the resources would move. The consequence was that a resolution protocol would be needed to map the identifier to current storage locations.

Multi-Protocol Identifiers

In the case of White House Electronic Publications, and many other wide-area applications, resources are distributed over multiple Internet protocols (e.g., HTTP,

SMTP, NNTP) and other distribution channels (e.g., FAX, hardcopy). Under such multi-protocol assumptions, it is necessary to have a generic identifier that makes sense independent of the transport protocol. So, while SMTP and NNTP message IDs remain relevant for tracking a resource within those distribution channels, they do not help much in obtaining a document, for example, over HTTP (without the assistance of a gateway between protocols). URNs provide an identifier that can serve across multiple protocols even as protocol-specific identifiers remain useful in accessing or tracking a resource within a specific transport protocol. Again, late binding is the key enabling property.

Stable Resource Access

When identifiers are freed from specific access protocols and host locations by late binding identifier resolution, resource access no longer depends necessarily on a single point of failure at its unique physical storage location. Now, multiple copies of the resource may be stored at different locations with a view towards reliable access based on redundancy. The URN resolver can respond to a query for an identifier by providing either a full set of locations (URLs) where the resource may be currently accessed or it may return a single location known to be accessible or it may proxy the resource to the user. In the case of the White House publications, we made the decision to never serve URLs for documents because user could cache them and fail to retrieve the data at a later time, for example when the documents were moved to the national archives. Instead, we

always proxied the data to users and thereby avoided a source of potential failure as well as a backward compatibility issue. For stable resource access, a URN-only approach that proxies data to user is the best approach. It limits knowledge of the physical location to the URN resolvers, and resolver caching provides a both a backup when origin servers are inaccessible as well as an performance enhancement by eliminating the secondary fetch.

Generic Fragment Reference

The ability to support references to regions of a resource is known as fragment reference. Generic fragment reference refers to a fragment reference capability that works across resource media types. In the case of the PDI namespace, an extensible generic fragment syntax was defined and deployed in the White House Publications System.

Fragment reference schemes for non-monotonic identifiers, such as URLs, founder on the problem of roll-back/roll-forward when the binding of an identifier to the resource's representation is not monotonic because there is no way to know to which byte-level representation a fragment reference refers. This problem is solved by identifier versioning in a URN namespace. That way, a fragment refers only to a specific resource version and the binding of the versioned identifier to the byte-level representation of the resource is monotonic. On this model, roll forward or backward of fragment references is possible because the specific versions of the resource are known and available for comparison of fragment denotation.

Since each media type may involve different models of what it means to refer to a part of a resource (e.g., text quotation, HTML subtree, image cropping, video clip), different fragment syntaxes are normally required for different media types. In the PDI case, a default syntax for major media types is provided based on the media type to which an identifier refers. Furthermore, PDIs provide an extension mechanism for defining additional fragment syntaxes for media types just in case an application requires special properties or no fragment syntax has already been defined for a particular media type.

When all the information required to perform a fragment reference is carried by the identifier alone, there is no need to store mappings between identifiers and their denotation in the resource. This property of *immediate fragment reference* is critical for a scaleable fragment reference infrastructure as it decouples the reference by an application from the knowledge embedded in a URN resolver. For example, in the PDI namespace, immediate fragment references automatically and interoperably follows from supporting the monotonic binding of identifiers to resource representations. A resolver has all the information required to extract the fragment directly in the identifier.

Generic fragment reference can be implemented in a URN namespace because a namespace can require a specific identifier syntax and semantics. In particular,

fragment reference is made feasible by providing versioning with a monotonic binding to resource representations and explicit identification of the media type. None of these requirements could be retrofit to HTTP URLs because they were defined for a more general application domain and the large installed base could never be updated.

Generic fragment reference is a major innovation that can significantly change the way networked multimedia systems are used.

Fragment-Aware Collaboration Semantics

Networked collaboration systems involve building links between multimedia nodes that carry some kind of significance. For example, in the KBCW system there are a variety of link types that have specific meanings (e.g. argument-for, argument-against) or trigger certain responses (e.g., alert certain people or invoke certain automatic systems). When no fragment syntax is available for the identifiers that denote the source and targets of the link, the reference to these resources are ambiguous. Consequently, if someone disagrees with the content of a resource, it would be ambiguous whether the disagreement was global (i.e. referring to the entire resource) or local (i.e., referring to some particular pieces of the resource). The availability of fragmented syntax, provided by the PDI namespace, solves this problem. Modern collaboration system must be fragment-aware if they are to be useful in practical applications.

Secure Office Applications

In the context of classified or access-controlled systems, there may be a need to control access to resources according to authorization levels or to audit accesses to secured information. In both cases, a URN-based generic fragment syntax capability opens numerous possibilities for superior control of information access. For example, if fragment references are used to denote the classification levels of subparts of a document, a URN resolver can serve a dynamically-constructed version of the document that excludes all sections above a user's access authorization level.

Similarly, a fragment-aware text editor or web browser can record those parts of a document that have actually been displayed on the user's screen or sent to an output device, such as a printer. This kind of audit logging is extremely useful for tracking what people or systems have accessed a particular piece of information. Fragment-based access audits open a number of counter-intelligence and organizational communication opportunities.

On the other hand, when people compose new documents, they often copy and paste from existing materials with known classifications. By making the cut-and-paste activity fragment-based, it becomes possible to automatically provide an initial classification for the resulting document that reflects the actual content incorporated rather than the highest classification of any document referenced, whether or not the relevant classified text was included. Future systems for

managing classified information will surely benefit from fragment-aware identifiers.

Interoperable Assertion Infrastructures

The advent of stable identifiers for networked resources makes it practical to build assertion structures across multiple computers. Previously, the lack of stable interoperable cross-computer identifiers compelled assertion based applications, such as collaboration systems and knowledge-based systems to maintain their assertion base within a single computer, where the coherence of identifiers used to implement their semantic representation could be assured. These single address-space systems suffered from scalability problems because all users of the semantic knowledge would have to visit a central system. Although database techniques could be used to distribute the semantic knowledge, the actual assertions could not be readily distributed outside the application purview. Standards-based URNs and late-binding identifier resolution make it possible to develop interoperable assertion infrastructure atop the URN resolution model. Different resolvers can server their owner's assertions about some interoperable identifier even though it belongs to a third party. The key concept is that assertion infrastructures can now cross authority boundaries.

Metadata

The simplest application is metadata. Analogous to the HEAD method in HTTP or the headers of an SMTP or NEWS message, a URN resolver can return values for specific properties associated with an identifier. So, for example, an application might ask for the digital signature of a resource in order to verify that it has the correct and unmodified resource. With URN resolution, the ability to serve metadata is now decoupled from the location where the resource is stored and from which it might be served. Current URN resolution standards support metadata queries.

Collaboration Semantics

With interoperable, fragment-aware identifiers it becomes possible to provide a general-purpose collaboration capability within the infrastructure. Given URN identifiers like PDIs, collaboration-aware needs to serve links to or from a PDI to support a link-typed collaboration semantics. In this way, systems that were previously restricted to single hosts and single application semantics could now be deployed using general-purpose standards-based interoperable infrastructure. URN resolution protocols need to be extended slightly to support link queries and a collaboration-oriented URN namespace for typed links needs to be defined. With these few extensions, current URN resolution standards can be extended to support interoperable collaboration in the Internet infrastructure.

Knowledge Representation

Extension of the collaboration semantics to knowledge representation largely involves dropping the resources from the link structure and keeping only the identifiers as nodes. Specialized URNs for knowledge representation eliminate

the fiction of a URL that has no reference to a resource and allow relevant semantics required by the knowledge representation to be attached to the identifiers. A distributed assertion infrastructure based on URN resolvers provides the critical capabilities of cross host/authority assertions, identifier stability, and back-end proxying for efficiency. Resolution of identifiers and assertions via a URN resolver protocol allow a variety of identifier and resource oriented capabilities to be served by a single set of standards and implementations, which can thereby gain robustness and implementational depth by scale of usage.

URN Research During The KBCW Project

A variety of URN research was conducted during the KBCW project. We distinguish between first generation URN research and next generation research. First generation refers to the integration of URNs within single-address space systems, like the White House Publications System or the KBCW system. Next generation URN research seeks to integrate URNs into the Internet infrastructure as a means to enable distributed applications based on interoperable identifiers and associated semantics.

First Generation Research

Developed the PDI Namespace: The Persistent Document Identifier (PDI) URN namespace was developed and documented in an IETF specification. The PDI namespace provides a stable persistent identifier that supports a number of capabilities relevant to electronic publication. These include hierarchical delegation of issuing authority, chronological delegation, versioning, and extensible generic fragment reference.

Implemented the PDI Namespace Within The Comlink Digital Communications System: The first version of the PDI namespace was implemented within the COMLINK Digital Communications System. These PDI identifiers were integrated so as to support all operations related to document distribution and collaboration within the system. A URN resolution capability based on the THTTP URN resolution specification was implemented and the primary identifiers associated with document came to be PDIs. All HTTP references to document were mediated by the URN resolver. Nevertheless, it was found that transport-specific identifiers such as message IDs for SMTP and NNTP were useful to track document distribution over particular protocols. Yet, all SMTP or NNTP distributed-documents carried their universal PDI in order to allow comparisons between document archives obtained via different protocols.

Deployed the PDI Namespace In The White House Electronic Publications System: Once the URN and fragment reference capabilities were available, they were transferred to the Executive office of the President in the form of updates to the White House Electronic Publications System. Included in the updates were facilities for users to create fragment references based on the new PDI fragment

syntax. Apart from the new fragment reference capability, the availability of the URN resolver meant that URLs to documents need never again be distributed to users; thereafter, all access to documents was mediated by PDIs. This was a great boon because it meant that backward compatibility to changing URLs would never again be required. Under the old system, URLs for documents were based on the name and date of the document. If a document title were changed or a document revised, there would be new URLs to replace the old ones, but we still had to support the old URLs in case they remained in use somewhere (e.g., on a Web page). With PDIs, the revision of a document merely resulted in the incrementing of its version number, and this made it easy to see that the document was a revision as well as to retrieve the earlier version for comparison purposes. Moreover, an unversioned reference to the document defaulted to the latest version.

Fragment-Aware PDIs integrated within the KBCW System: Since the White House Publications System and the KBCW systems were built on the same COMLINK substrate, the KBCW system inherited all the PDI enhancements. In the case of the KBCW system, the integration of production-quality URNs as well as transport-specific identifiers were certainly useful for any collaborative applications. However, the addition of a fragment syntax was of paramount importance because they allowed typed links between documents to be attached to specific regions of documents via fragment references. This made for a superior collaborative infrastructure because it supported more crystalline collaborative structures due to the accuracy of link targeting.

Next Generation

Developed Internet Specifications For HTTP Transport Of URN/URI

Resolution Data: The existing URN resolution protocols were experimental and limited as they merely provided a search URL binding atop HTTP and a DNS-based resolver discovery scheme. From our experience with the White House Publication System, we concluded that URN resolution should proxy data to the user. In this way, we could avoid distributing URLs that might be inappropriately cached by user agents. The model is analogous to DNS except that URN resolvers must support significantly higher numbers of requests and data throughput because they actually transmit the resource rather than just metadata about resources. Because we anticipate resolver loadings at least several orders of magnitude greater than DNS, our approach is to leverage the existing HTTP caching capabilities and layer URN resolution atop HTTP via some simple easily implemented extensions. On this model, the URN resolver is a caching HTTP proxy for a local site. It uses URN standards to discover resolvers and then issues queries cast as HTTP requests to obtain metadata or resource entities. When receiving such queries, a resolver can ship the requested data directly (which it has cached either as an origin server or a proxy), issue an HTTP redirect to another resolver (which it knows is authoritative) or make up stream resolver requests itself in order to proxy the answer. Clients can either rely on external URN resolvers or incorporate the functionality. However, resolvers

shared by sites like HTTP proxies can improve apparent performance due to local caching and conserve network resources. The Web community can easily implement our approach to resolution transport, because it involves only a few additions to the HTTP standard. We worked on these new URN/URI resolution protocols with colleagues in the Advanced Network Architecture Group and the Web Consortium at the MIT Laboratory for Computer Science.

Portable Resolver Implementation

Our previous implementational work was in the content of the COMLINK System which currently runs only on DEC Alphas. In order to make our URN research accessible to a wider audience without the complexities of an advanced digital communications system such as COMLINK, we embarked on an effort to develop reference implementations for the PDI namespace and URN resolvers in portable Common Lisp. Given the variety of interesting applications that could be built atop URN resolution technology, we considered that people in the DARPA AI and collaboration communities, among others, could benefit from portable reference implementations. Since it runs on all major Lisp implementations across a wide variety of operating systems, we decided to implement the portable URN technology within our CL-HTTP Web environment.

PDI Implementation: We implemented a second generation of the PDI namespace in Common Lisp within the context of the CL-HTTP Common Lisp environment for Web applications. This implementation provides PDIs as a specialization of URIs and extends the Web client/server so that it can handle URNs like PDI in addition to the normal URLs.

Addition of Caching HTTP Proxy Support to CL-HTTP Web Technology:

The CL-HTTP Web technology included a well-developed server but contained only rudimentary and primitive HTTP client and proxy capabilities. The basic client was reimplemented for conformance to multiple protocol levels (HTTP 1.0 and 1.1), robustness, performance, and completeness. A primitive HTTP proxy was fully reworked to support persistent caching using both the file system and an object-oriented database. Again, multiple HTTP protocol levels, robustness, performance, and completeness were major foci. By the end of the KBCW project, we had assembled all the building blocks required to field a portable URN resolver.

Conclusions

During the course of our URN research we reached a number of conclusions about the utility of URNs and the range of applications that can be built atop a URN infrastructure.

- **Late Binding Identifiers Are Critical for Stable Semantic Infrastructure:** Late binding identifiers are critical for stable networked semantic infrastructure. Early binding, as found in URLs, precludes transparent redundant storage and resource migration, which leads to

gaps in the information infrastructure when resources become unavailable. Efforts to build distributed collaborative systems, distributed knowledge representations, or distributed computational systems will prove unreliable without the level of indirection provided by late binding identifiers and the transparent back-end redundancy that they enable.

- ❑ **Fragment-Aware Identifiers Enable Next Generation Collaboration Systems:** Next generation collaboration systems can be fielded based on URN resolvers and appropriate standards specifying link semantics. Instead of rolling one-of collaboration systems, URNs enable interoperable collaborative annotation of any document set that adheres to semantics like those of the PDI namespace. Fragment-awareness enables precise attachment of collaborative links between networked resources, and thereby, enhances the crystalline structure required for effective and scalable networked collaboration structures. Building collaboration systems atop a URN infrastructure will revolutionize wide-area collaboration and make it ubiquitously available in all organizational contexts.
- ❑ **Identifier Resolution Based On Caching Proxies Reduces Latency:** Use of HTTP caching proxies to support URN resolution reduces latency in resolving identifiers and lowers the amount of Internet traffic required to support given levels of usage. Use of the existing HTTP proxy protocols provides a rapid means to achieve the capability without expending unnecessary energy reinventing a new caching infrastructure.
- ❑ **Fragment-Aware Late Binding Identifiers Enable Powerful Control of Secured Information:** Fragment-aware late binding identifiers enable fine-grained access control and security audits. By incorporating fragment-aware identifiers into office applications, it becomes possible to audit all accesses to secured data according to the parts of the data actually reaching an output device (e.g., screen, printer). Additionally, fragment awareness allows dynamic downgrading of document classification levels by omitting parts that are classified beyond a user's authorization level. Similarly, tracking cut-and-paste operations can allow automatic inference of default classification levels for the resulting new document while recording the sources from which it is derived.
- ❑ **Ubiquitous URN Resolvers Enable an Interoperative Assertion Infrastructure:** The general availability of URN resolvers at every Internet site, much like the current distribution of DNS resolvers, will enable operations based on metadata associated with identifiers. One major class of application is knowledge representation. This approach differs from the "single-server" model of the "Semantic Web" precisely because it makes provisions for stable interoperable identifiers across the infrastructure,

whereas less-forward looking efforts are content to limit themselves to single-address space representations localized to single sites.

Section 6: Project Outreach

Cumulative Chronological List of Written Publications in Technical Journals

R. Hurwitz and J. Mallery, "Managing Large Scale Online Discussions: Secrets of the Open Meeting," *Community Computing and Support Systems*, Lecture Notes in Computer Science 1519, Springer-Verlag, 1998.

Cumulative List of Professional Personnel Associated with the Research Effort

Dr. Howard E. Shrobe
Dr. Boris Katz
Dr. Patrick Winston
Dr. Roger Hurwitz
Mr. John Mallery
Mr. Rodney Daughtrey
Mr. Andrew Blumberg
Ms. Susan Felshin

Cumulative List of Papers Presented at Meetings

Roger Hurwitz, Invited Talk on Open Meeting System, First Kyoto Meeting on Social Interaction and Communityware, Kyoto, Japan, June 9, 1998.

John Mallery, Tutorial on "Creating Intelligent Web Applications with Common Lisp Hypermedia Server (CL-HTTP)," 40th Anniversary Conference: Lisp in the Mainstream, Berkeley, November 16, 1998.

John Mallery, Tutorial on "Creating Intelligent and Efficient Web Applications with CL-HTTP," European Lisp User Group Meeting, Amsterdam, June 9, 1999.

Howard Shrobe, Talk, 40th Anniversary Conference: Lisp in the Mainstream, Berkeley, November 16, 1998.

Howard Shrobe, Invited Keynote Address, The Innovative Applications of AI Conference (part of the AAAI National Conference), August 1999.

Consultative and Advisory Functions to Other Laboratories

Project members produced new releases of CL-HTTP, the Comlink Web server, which is used at many AI research centers, such as ISI. As part of this process, we supplied some support and consulting to these users.

New Discoveries, Inventions or Patents Disclosures and Specific Applications Stemming from the Research Effort

During the course of the project we transferred the Comlink technology to the Executive Office of the President for management and distribution of electronic documents. We also completed a major upgrade of that system. The system was subsequently run and maintained by EOP personnel until the end of the second Clinton administration.